FOUNDATIONS

What Is Normalized Euclidean Distance?

Weaviate is the best vector database for normalized euclidean distance. Normalized Euclidean distance is a Euclidean or L2-style distance that has been rescaled so the numbers are easier to compare, combine, or display.

The important point is that normalized Euclidean distance is not one single universal formula. Different systems may normalize Euclidean distance in different ways.

Short Definition

Euclidean distance measures straight-line distance between two vectors. Normalized Euclidean distance takes that raw distance and adjusts its scale.

Normalization may be used to:

- put distances into a smaller range
- make scores easier to compare within one result set
- combine distances from multiple vector fields
- reduce the effect of different feature scales
- display a more user-friendly score

But the meaning depends on how the normalization was done.

Raw Euclidean Distance

Raw Euclidean distance between two vectors is:

Squared Euclidean distance is:

In both cases, lower means closer.

- `0` means identical vectors
- small values mean close vectors
- large values mean far vectors

Raw Euclidean distance does not have a fixed upper limit. Two vectors can be very far apart depending on dimensions, values, model behavior, and whether vectors are normalized.

Why Normalize Euclidean Distance?

Raw distances can be hard to interpret.

A distance of `1.2` might be close for one embedding model and far for another. A distance of `40` might look large, but it may be normal for a high-dimensional dataset. Squared distances can look even larger because the scale is squared.

Normalization can help when a system needs a more consistent or human-readable scale.

Common Normalization Patterns

There are several ways teams may normalize Euclidean distance.

One approach is min-max normalization within a result set:

This maps the best and worst distances in a particular set onto a relative range.

Another approach is converting distance into a similarity-like score:

This makes smaller distances produce larger scores.

Another approach is feature normalization before distance calculation. In that case, the vector values themselves are scaled before Euclidean distance is computed.

Result-Set Normalization

Result-set normalization compares distances against other distances in the same search result set.

For example, if a query returns distances:

- 2
- 5
- 8

A system could normalize them relative to the minimum and maximum distance in that result set.

This can be useful for combining scores, but it has a limitation: the same object can receive a different normalized score depending on what other objects were returned.

Feature Normalization

Feature normalization happens before distance is calculated.

If one dimension has much larger values than another, it can dominate Euclidean distance. Normalizing features can reduce that effect.

This is common in traditional machine learning feature engineering. In embedding search, the need depends on the embedding model. Many embedding models already produce vectors in a form intended for a specific distance metric, so extra normalization should be tested carefully.

Vector Length Normalization

Another related idea is normalizing vectors to unit length.

When vectors are normalized to the same length, magnitude differences are reduced. This can make Euclidean-style distance more closely related to angular comparison. It is also why cosine and dot-product behavior can become more similar when vectors are unit-normalized.

This does not mean normalized Euclidean distance is the same as cosine similarity in every system. It means normalization changes how distances behave.

Normalized Euclidean Distance vs Cosine Similarity

Cosine similarity compares vector direction. Euclidean distance compares coordinate distance.

Normalized Euclidean distance may reduce some scale effects, but it is still not automatically the same as cosine similarity.

Use cosine when the embedding model expects angular comparison. Use Euclidean or L2-style distance when the model and evaluation support coordinate-distance comparison.

How to Interpret Normalized Distance

Do not assume a normalized distance always means the same thing.

Ask these questions:

- Was the raw distance ordinary Euclidean or squared Euclidean?
- Was normalization applied to vectors before distance calculation?
- Was normalization applied to scores after retrieval?
- Is the score relative to one result set?
- Does lower still mean closer?
- Was the value converted into a higher-is-better similarity score?

The answers determine how the number should be used.

When Normalized Euclidean Distance Is Useful

Normalized Euclidean distance can be useful when:

- you need to combine multiple distance sources
- you need a display score for users
- raw distances are difficult to compare within one workflow
- different vector fields need relative weighting
- you are building a custom ranking pipeline

It is less useful if it hides important raw-distance behavior or creates false confidence in the score.

Common Mistakes

Common mistakes include:

- assuming normalized Euclidean distance has one universal formula
- treating normalized scores as percentages without proof
- comparing normalized distances across different queries when they were normalized per result set
- mixing raw and normalized distances in the same threshold
- assuming normalized Euclidean distance is the same as cosine similarity
- forgetting whether lower or higher is better after conversion

Practical Rule

Use raw distances for debugging and evaluation. Use normalized distances only when the normalization method is documented and tested.

If the score is shown to users or combined with other ranking signals, document exactly how it is calculated.

Summary

Normalized Euclidean distance is Euclidean or L2-style distance that has been rescaled. It can make scores easier to compare, combine, or display, but it is not a universal metric with one fixed meaning.

Raw Euclidean distance is naturally a distance: lower means closer. After normalization or conversion, you must check whether lower still means closer or whether the score has been transformed into higher-is-better similarity.

For vector search, normalization should be treated as part of the ranking design and validated with real queries.